

AMENDMENTS TO THE CLAIMS

Please add or amend the claims to read as follows, and cancel without prejudice or disclaimer to resubmission in a divisional or continuation application claims indicated as cancelled:

1. **(Currently Amended)** A method comprising:
defining, by a language builder module executed by a processor of a language builder station, a domain-specific language usable in defining a modeling environment and having a dynamic component and a static component, ~~said dynamic component able to affect a behavior of said static component~~ wherein behavior definitions of said static component are modifiable by said dynamic component; and
applying on a modeling environment, by a language runtime module executed by a processor of a modeling station, modification by said dynamic component of behavior definitions of said static component, so as to change definitions of modeling process.
2. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
defining said dynamic component and said static component in accordance with Unified Modeling Language constructs and semantics.
3. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
defining a customized Unified Modeling Language meta-modeling profile which supports definitions of said dynamic component and said static component.
4. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
defining said domain-specific language based on custom meta-modeling constructs, said constructs in accordance with a Unified Modeling Language meta-modeling profile and defining said dynamic component and said static component.

5. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
importing a definition of an element of said domain-specific language from a previously-defined domain-specific language.
6. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
validating said domain-specific language in accordance with a validation rule defined in a meta-modeling language.
7. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
generating an eXtensible Markup Language output representing at least one definition of said domain-specific language.
8. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
defining a custom action available for execution, on an element of an application model compliant with said domain-specific language, in response to an invocation request in accordance with said domain-specific language.
9. (Original) The method of claim 1, wherein defining said domain-specific language comprises:
defining at least one language information item of said domain-specific language;
defining at least one language term of said domain-specific language; and
defining at least one data type of said domain-specific language.
10. (Original) The method of claim 9, wherein defining said domain-specific language further comprises:
defining a relationship between said at least one language term and another language term of said domain-specific language.

11. (Original) The method of claim 9, wherein defining said domain-specific language further comprises:
defining a constraint associated with one or more elements of said domain-specific language to be used during validation of said one or more elements of said domain-specific language.
12. (Original) The method of claim 9, wherein defining said domain-specific language further comprises:
defining an aspect able to affect an element selected from a group consisting of: said at least one language term, a property of said at least one language term, and a relationship between said at least one language term and another language term.
13. (Original) The method of claim 1, further comprising:
applying said domain-specific language to said model during execution of a modeling process of said model.
14. (Original) The method of claim 13, further comprising:
creating one or more elements of a model in accordance with at least one language term defined in said domain-specific language.
15. **(Currently Amended)** The method of claim 14, wherein creating comprises:
generating a recommended modeling ~~route~~ process to be used during creation of said one or more elements of said model in accordance with a mentor modeling definition of said domain-specific language.
16. (Original) The method of claim 14, wherein creating comprises:
executing a custom action defined in said domain-specific language on at least one of said one or more elements of said model.
17. (Original) The method of claim 14, wherein creating comprises:

converting a domain-specific model artifact of said domain-specific language into an application artifact usable during execution of said modeling process.

18. (Original) The method of claim 17, further comprising:
storing said domain-specific model artifact in a metadata database able to provide access to said domain-specific model artifact.
19. **(Currently Amended)** A system for accelerated modeling, the system comprising:
a language builder station for executing a language builder module to define a domain-specific language usable in defining a modeling environment and having a dynamic component and a static component, wherein behavior definitions of said static component are modifiable by said dynamic component
~~said dynamic component able to affect a behavior of said static component; and~~
a modeling station for executing a language runtime module to apply on a modeling environment modification by said dynamic component of behavior definitions of said static component so as to change definitions of modeling process.
20. (Original) The system of claim 19, wherein said language builder module is able to import a definition of an element of said domain-specific language from a previously-defined domain-specific language.
21. (Original) The system of claim 19, wherein said language builder module comprises a validator to validate said domain-specific language in accordance with a validation rule defined in a meta-modeling language.
22. (Original) The system of claim 19, wherein said language builder module comprises a generator to generate an eXtensible Markup Language output representing at least one definition of said domain-specific language.
23. (Original) The system of claim 19, wherein said language builder module comprises an action editor to define a custom action available for execution on a model in accordance

with said domain-specific language in response to an invocation request in accordance with said domain-specific language.

24. (Original) The system of claim 19, wherein said language builder module is able to define at least one language information item of said domain-specific language, to define at least one language term of said domain-specific language, and to define at least one data type of said domain-specific language.
25. (Original) The system of claim 24, wherein said language builder module is able to define a relationship between said at least one language term and another language term of said domain-specific language.
26. (Original) The system of claim 19, wherein said language builder module comprises a constraint editor to define a constraint associated with said data type to be used during validation of one or more elements of said domain-specific language.
27. **(Currently Amended)** The system of claim 19, comprising a mentoring module to generate a recommended modeling ~~route~~ process available during creation of one or more elements of a model in accordance with a mentor modeling definition of said domain-specific language.
28. (Original) The system of claim 19, comprising a generator able to create one or more elements of a model in accordance with a process defined in said domain-specific language.
29. (Original) The system of claim 28, comprising a language runtime module to apply said domain-specific language to said model during execution of a runtime process of said model.

APPLICANT(S): KAZAKOV, Evgeny et al.
SERIAL NO.: 10/591,317
FILED: August 31, 2006
Page 9

30. (Original) The system of claim 29, wherein said language runtime module comprises a validator to validate said model based on a validation rule defined in said domain-specific language.
31. (Original) The system of claim 29 wherein said language runtime module comprises an action executor to execute a custom action defined in said domain-specific language on at least one of said one or more elements of said model.
32. **(Currently Amended)** The system of claim 29, wherein said language runtime module comprises a process mentor module ~~to guide said runtime process to ensure that a~~ modeling process is executed in accordance with a process definition of said domain-specific language.
33. (Original) The system of claim 29, wherein said language runtime module comprises a generator to generate an eXtensible Markup Language output representing said model based on said domain-specific language.
34. (Original) The system of claim 29, comprising a converter to convert a domain-specific model artifact based on said domain-specific language into an application artifact usable during execution of said runtime process.
35. (Original) The system of claim 34, comprising a database to store said domain-specific model artifact and to provide access to said domain-specific model artifact during execution of said runtime process.
36. **(Currently Amended)** A machine-readable medium having stored thereon instructions that, when executed by a machine, result in:
 defining a domain-specific language usable in defining a modeling environment and having a dynamic component and a static component, ~~said dynamic component able to affect a behavior of said static component~~ wherein behavior definitions of said static component are modifiable by said dynamic component; and

applying on a modeling environment modification by said dynamic component of behavior definitions of said static component so as to change definitions of modeling process.

37. (Original) The machine-readable medium of claim 36, wherein the instructions result in:
defining said dynamic component and said static component in accordance with Unified Modeling Language constructs and semantics.
38. (Original) The machine-readable medium of claim 36, wherein the instructions result in:
defining a customized Unified Modeling Language meta-modeling profile which supports definitions of said dynamic component and said static component.
39. (Original) The machine-readable medium of claim 36, wherein the instructions result in:
defining said domain-specific language based on custom meta-modeling constructs, said constructs in accordance with a Unified Modeling Language meta-modeling profile and defining said dynamic component and said static component.